

APLIKASI PIRANTI LUNAK PENGANALISIS SINTAKSIS KALIMAT BAHASA INDONESIA

Sablin Yusuf

Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Binus University
Jln. K.H. Syahdan No. 9, Kemanggis/Palmerah, Jakarta 11480
sablin@binus.edu

ABSTRACT

Indonesian has a vast variety in structures and sentence types that brings about difficulty in using the language correctly and accurately. This research aims for producing a software application for syntax analysis and error checking for Indonesian sentences. The method used in developing the application is rapid prototyping method which consists of several phases: system requirement analysis, design, coding, implementation and evaluation while the method used in syntax analysis process is regular expression and predictive parsing method. Despite some existing limitations, the application produced is able to check the syntax error of various Indonesian sentence types.

Keywords: *syntax analyzation, Indonesian Language, application software.*

ABSTRAK

Bahasa Indonesiamemiliki ragam yang luas dalam struktur dan jenis kalimatnya. Kondisi ini menyebabkan kesulitan dalam berbahasa Indonesiadengan tepat dan benar. Penelitian ini bertujuan menghasilkan aplikasi untuk menganalisis sintaksis dan memeriksa kesalahan kalimat berbahasa Indonesia. Metode yang digunakan dalam pengembangan aplikasi ini adalah rapid prototyping yang terdiri dari: analisis kebutuhan sistem, perancangan sistem, implementasi dan evaluasi sistem Sedangkan metode yang digunakan dalam proses analisis sintaksis adalah regular expression dan predictive parser. Aplikasi yang dihasilkan mampu memeriksa kesalahan sintaksis berbagai bentuk kalimat berbahasa Indonesia.

Kata kunci: *analisis sintaksis, Bahasa Indonesia, aplikasi piranti lunak.*

PENDAHULUAN

Manusia menggunakan bahasa sebagai alat berkomunikasi, baik itu secara lisan maupun tertulis. Penggunaan bahasa yang benar mengurangi kemungkinan pembaca untuk mengira-ngira maksud penulis, sehingga lebih menjamin pemahaman penulisan sesuai dengan pesan yang hendak disampaikan oleh penulis. Dalam konteks bahasa manusia, sintaksis merupakan pengetahuan mengenai bagaimana menggabungkan kata-kata menjadi kalimat yang benar.

Salah satu ciri khas yang dihadapi bahasa manusia adalah ketaksaan (*ambiguity*), yaitu suatu deretan kata dalam satu kalimat dapat memiliki lebih dari satu fungsi atau makna. Dengan demikian, kalimat tersebut juga memiliki lebih dari satu makna. Akibatnya, jika kita melakukan analisis sintaksis terhadap kalimat tersebut, kita akan mendapatkan lebih dari satu pohon sintaksis pula. Hasil dari analisis sintaksis sangat diperlukan untuk berbagai aplikasi, seperti mesin penerjemah, *information retrieval*, dan sebagainya.

Bahasa Indonesia relatif lebih sederhana apabila dibandingkan dengan Bahasa Inggris. Dalam Bahasa Indonesia tidak mengenal *tenses*, dan pembentukan bunyi atau pembacaan katanya mempunyai aturan yang jelas dari huruf-huruf yang membentuknya. Namun pengaruh dari ratusan bahasa daerah dan bahasa asing, terutama Bahasa Arab dan Inggris, cukup berpengaruh banyak terhadap penggunaan Bahasa Indonesia (Arman, 2005).

Beberapa usaha atau penelitian pernah dilakukan untuk proses analisis sintaksis Bahasa Indonesia, antara lain analisis sintaksis Bahasa Indonesia dengan menggunakan metode *Generalize LR-Parsing*, (Mandala et al, 2004). Namun lingkup penelitian dibatasi hanya untuk beberapa pola kalimat dasar Bahasa Indonesia. Penelitian ini bertujuan untuk menghasilkan suatu piranti lunak penganalisis sintaksis kalimat Bahasa Indonesia, untuk memeriksa dan menemukan kesalahan-kesalahan sintaksis dalam penulisan kalimat berbahasa Indonesia sehingga dapat membantu pengguna dalam menghasilkan suatu dokumen dengan tingkat kesalahan sintaksis yang rendah.

Proses analisis sintaksis dilakukan dengan metode *predictive parsing* dengan memanfaatkan *regular expression* untuk menguraikan kalimat yang akan diperiksa (Aho, 2007). *Regular expression* merupakan suatu bentuk notasi yang dapat digunakan untuk mengolah teks, termasuk di dalamnya mendeskripsikan dan memisahkan teks (Raharjo, 2001).

Beberapa manfaat yang didapatkan dari penelitian ini adalah (1) memudahkan pengguna untuk mengubah dan memeriksa kesalahan penulisan secara manual; (2) memberikan informasi kepada pengguna, apakah sintaksis sebuah kalimat sudah sesuai dengan kaidah tata bahasa atau belum; (3) sebagai dasar dalam membangun sebuah sistem pemeriksa Bahasa Indonesia secara lengkap mencakup sintaksis dan semantiknya.

Karena sangat luasnya ragam struktur bahasa Indonesia, maka ruang lingkup penelitian ini dibatasi sebagai berikut: (1) pemeriksaan yang dilakukan adalah pemeriksaan secara sintaksis berdasarkan kaidah tata bahasa (Sakri, 1994, p.8); (2) jenis kalimat yang dapat diproses adalah kalimat sederhana, kalimat bertingkat, dan kalimat kompleks; (3) file yang diproses berupa *Rich Text Format* (.RTF); (4) pemeriksaan dilakukan pada penggunaan tanda baca yang terdiri dari tanda titik (.), tanda koma (,), tanda seru (!), tanda tanya (?), dan tulisan yang terdiri dari abjad (a sampai dengan z), angka (0 sampai dengan 9); (5) pengguna dari sistem ini adalah peneliti bahasa yang terbiasa menggunakan komputer sebagai alat bantu.

METODE

Metode yang digunakan dalam penelitian ini adalah metode *rapid prototyping* yang di dalamnya terdapat tahapan-tahapan sebagai berikut: analisis kebutuhan, perancangan dan pembuatan prototipe, pengujian dan evaluasi, dimana perulangan terhadap keseluruhan proses ini terus dilakukan hingga tujuan dari sistem penganalisis sintaksis kalimat Bahasa Indonesia dapat tercapai sesuai dengan kebutuhan (Pressman, 2001).

Analisis Kebutuhan

Tahapan ini dimulai dengan pengumpulan data untuk memenuhi kebutuhan sistem baik melalui diskusi dengan berbagai calon pengguna, serta mendalami pustaka dengan mempelajari teori-teori mengenai aturan dan kaidah tata bahasa Indonesia, cara pembentukan kalimat bahasa Indonesia, struktur *database* yang digunakan, teknik *parser* yang digunakan dalam suatu kompilator, tampilan antar muka yang digunakan dalam sistem, dan teori pendukung lain melalui artikel yang diperoleh dari internet, survei ke beberapa institusi yang melakukan penelitian teknologi bahasa.

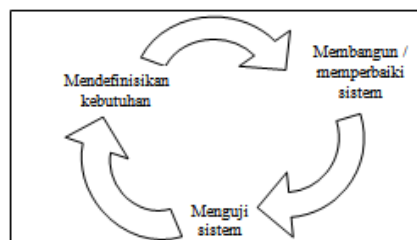
Perancangan dan Pembuatan Prototipe

Tahapan perancangan dilakukan dengan menyiapkan *flowchart*, *transition diagram*, rancangan *database* yang akan digunakan, tampilan layar sistem, dan rancangan spesifikasi proses. Proses ini dibarengi dengan proses *Quick design* yang berfokus pada representasi dari aspek-aspek piranti lunak yang terlihat oleh pelanggan atau pengguna. *Quick design* akan menuju pada pembuatan prototipe sistem.

Pengujian dan Evaluasi

Prototipe akan dievaluasi oleh pelanggan atau pengguna dan kebutuhannya akan disaring ulang untuk dikembangkan. Dalam tahapan ini Prototpe akan dievaluasi oleh pelanggan atau pengguna dan kebutuhannya akan disaring ulang untuk dikembangkan.

Untuk lebih jelasnya model *Rapid Prototyping*, dapat dilihat pada Gambar 1.



Gambar 1. Prototyping Model.

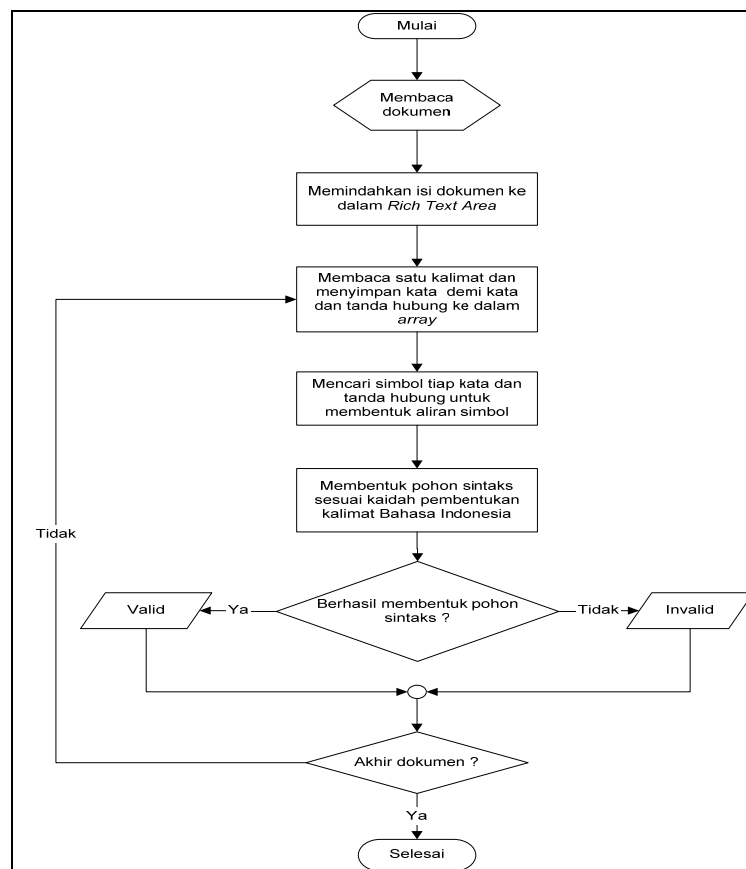
Prototyping model berawal dengan pengumpulan kebutuhan-kebutuhan. Pada tahap ini, pengembang piranti lunak dan pelanggan saling bertemu dan mendefinisikan tujuan dari piranti lunak yang ingin dibuat, mengidentifikasi kebutuhan apa saja yang sudah diketahui, dan batasan dari sistem. Setelah tahap ini dapat dihasilkan *quick design*. *Quick design* berfokus pada representasi dari aspek-aspek piranti lunak yang terlihat oleh pelanggan atau pengguna. *Quick*

design akan menuju pada pembuatan *prototype*. *Prototype* akan dievaluasi oleh pelanggan atau pengguna dan kebutuhannya akan disaring ulang untuk dikembangkan.

HASIL DAN PEMBAHASAN

Rancangan Umum Sistem

Rancangan umum sistem yang dihasilkan dapat dilihat pada Gambar 2. Saat sistem diberi masukan sebuah dokumen *Rich Text Format* (RTF) – semacam format teks yang mampu menampilkan atribut-atribut dokumen secara umum, sistem akan membaca seluruh isi dokumen dan memindahkannya ke dalam sebuah *Rich Text Area*.



Gambar 2. Bagan Alur Rancangan Umum Sistem.

Setelah itu, sistem akan memisahkan kata demi kata dan disimpan ke dalam sebuah *array*. Pemisahan berdasarkan spasi, dengan cara membaca karakter satu per satu. Sistem akan berhenti memisahkan kata apabila menemukan tanda baca yang menandakan akhir sebuah kalimat, seperti: tanda tanya (?), tanda titik (.), dan tanda seru (!).

Setelah sistem mendapatkan kumpulan kata-kata yang tersimpan dalam *array*, sistem akan mencari tiap kata tersebut ke dalam *database* untuk mendapatkan simbol tiap-tiap kata. Dalam sistem yang dibangun, hanya ada satu *entry* untuk suatu kata yang berguna mengatasi

adanya keambiguan dalam mengidentifikasi kata yang dimaksud. Penentuan jenis kata dilakukan dengan mencari jenis kata tersebut dalam Kamus Besar Bahasa Indonesia (KBBI) dan Kamus Elektronik Bahasa Indonesia (KEBI), jika ditemukan adanya sebuah kata dengan jenis yang berbeda, maka akan diambil jenis kata yang lebih umum dipakai dalam kalimat.

Jika kata tersebut tidak ditemukan dan kata tersebut terdiri dari bilangan dan tanda pengelompokan bilangan seperti titik (.), koma (,), dan persen (%), maka kata tersebut akan dinyatakan sebagai kata bilangan. Bila bukan, maka dianggap sebagai *identifier*, dalam hal ini sebagai kata benda. Dalam visualisasinya, kata-kata yang belum ada dalam *database* akan berwarna merah, dan yang sudah ada akan berwarna biru. Perbedaan atribut warna ini juga dimaksudkan agar pengguna dapat langsung mengetahui jika ada kata yang masih belum dikenal oleh sistem. Simbol-simbol yang didapat dari proses sebelumnya membentuk aliran simbol yang urutannya sesuai dengan kalimat masukan. Kumpulan simbol-simbol yang berurutan ini untuk selanjutnya disebut sebagai aliran simbol. Aliran simbol akan diteruskan ke sebuah modul yang memanfaatkan teknik rekursif untuk mulai memproses aliran simbol tersebut dan membentuk pohon sintaks. Dalam pembentukan pohon sintaksis ini digunakan *regular expression* untuk memeriksa apakah urutan simbol sesuai dengan aturan kaidah tata Bahasa Indonesia yang tersimpan dalam *database*.

Proses akan berlangsung hingga semua simbol selesai diproses atau terjadi keadaan dimana sistem tidak mampu menemukan kombinasi simbol yang benar dan cocok dengan kalimat masukan. Bila pohon sintaksis berhasil memproses semua simbol, maka sebuah variabel valid sebagai penanda akan terisi dengan nilai benar dan sebaliknya bila tidak berhasil, maka akan terisi dengan nilai salah. Setelah sistem selesai memproses satu kalimat hasil dari pemisahan di atas, maka akan diperiksa apakah kalimat tersebut merupakan kalimat terakhir atau tidak. Jika kalimat tersebut bukan merupakan kalimat terakhir, maka proses akan diulang untuk kalimat selanjutnya. Tetapi jika kalimat tersebut merupakan kalimat terakhir, maka aplikasi selesai untuk memproses semua kalimat masukan yang dimasukkan pengguna.

Dalam penelitian ini, pemeriksaan pola kalimat dilakukan berdasarkan pendekatan kaidah bangun frasa. Adapun aturan tata bahasa yang digunakan mengacu pada kaidah bangun frasa menurut Sakri (1994, p8a), antara lain:

M	→	(Ra) PB PK	(aturan 1)
PB	→	B	(aturan 2)
PK	→	Tokok(0,5) K (PB) (PD) ⁺	(aturan 3)

Kemudian masing-masing *token* akan dikonversikan ke dalam simbol yang bersesuaian seperti yang ditunjukkan pada Tabel 1.

Tabel 1
Tabel Konversi Token ke Simbol

Token	Simbol
M	M
Ra	R
PB	N
PK	V
B	B
Tokok	T
K	K
PD	P

Pemeriksaan kalimat ini menggunakan komponen *regular expression*, sehingga semua kaidah bangun frase dikonversikan ke dalam notasi *regular expression*.

Misalnya untuk aturan 1, notasi *regular expression*-nya menjadi:

< M > → ^R?[N][V]\$

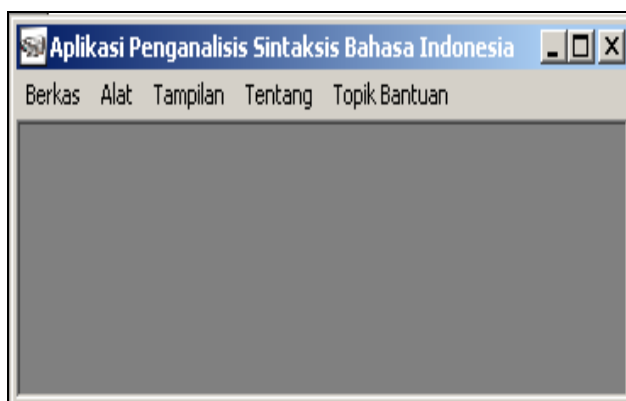
Proses pemeriksaan dilakukan dengan pengambilan kata per kata, sehingga tata bahasa tersebut perlu diurai agar memungkinkan pemeriksaan yang benar secara parsial.

Misalnya untuk *regular expression* di atas, akan diurai menjadi:

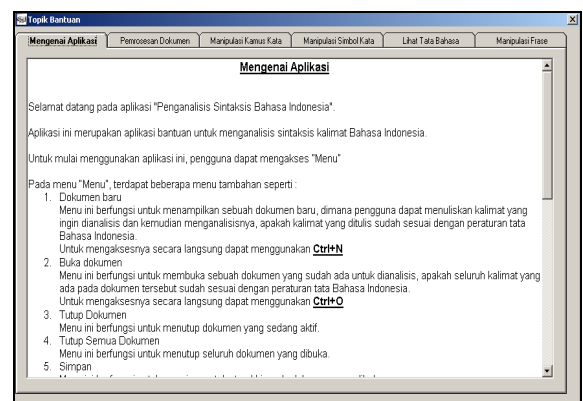
< M >	→	^R\$	(Parsial 1)
< M >	→	^R?[N]\$	(Parsial 2)
< M >	→	^R?[N][V]\$	(Parsial 3)

Implementasi

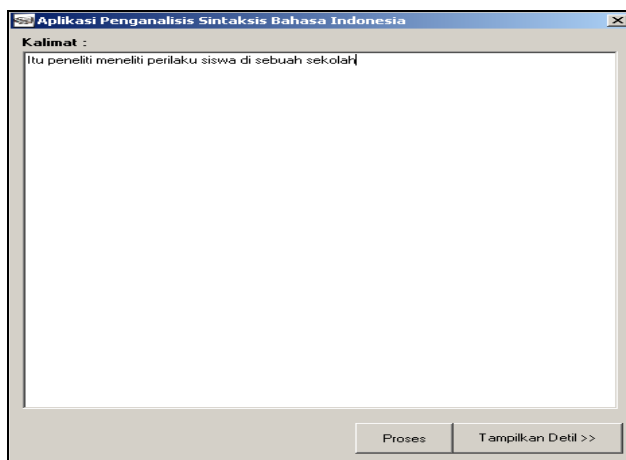
Gambar 3 s/d 8 merupakan beberapa contoh tampilan layar dari aplikasi yang dihasilkan.



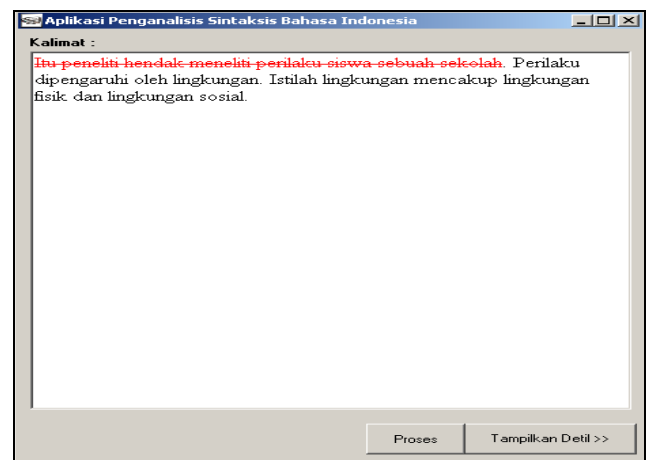
Gambar 3. Tampilan menu utama.



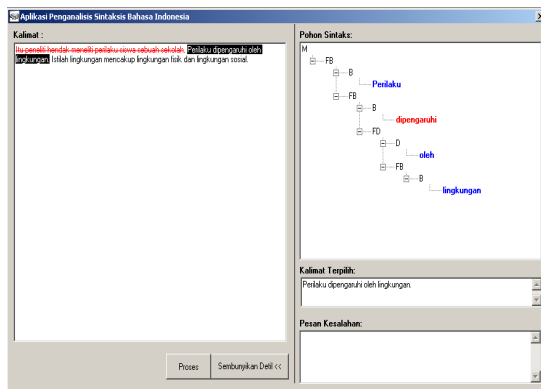
Gambar 4. Tampilan layar 'topik bantuan'.



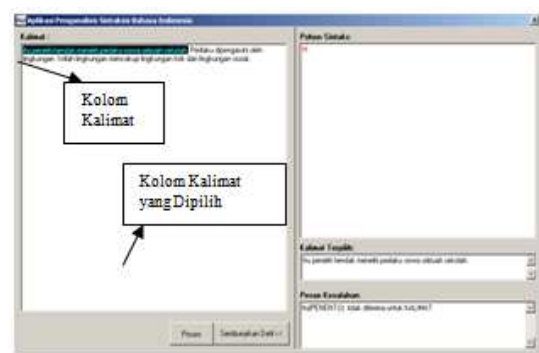
Gambar 5. Window untuk menuliskan kalimat.



Gambar 6. Hasil analisis kalimat yang berasal dari dokumen.



Gambar 7. Tampilan layar hasil analisis kalimat sesuai kaidah tata bahasa.



Gambar 8. Tampilan layar mengganti kalimat tidak sesuai kaidah tata bahasa.

Pembahasan

Evaluasi sistem dilakukan dengan melakukan pengujian pada jenis kalimat tunggal, kalimat bertingkat dan kalimat kompleks, selain itu evaluasi juga dilakukan baik itu pada kalimat yang sesuai ataupun kalimat yang tidak sesuai dengan kaidah tata bahasa yang dipakai. Tabel 2 dan 3 menampilkan hasil pengujian yang telah dilakukan. Untuk memeriksa kalimat Bahasa Indonesia secara semantik, pengembangan bukan terletak pada *engine* aplikasi, tetapi dengan melakukan penelitian yang mendalam mengenai penggolongan kata secara semantik sehingga dapat membentuk aturan kaidah tata bahasa yang mencakup semantik.

Secara keseluruhan, ada beberapa hal yang mampu dilakukan aplikasi ini. Pertama, secara garis besar sistem telah dapat berfungsi dengan baik untuk dapat memeriksa sintaksis kalimat Bahasa Indonesia dalam hal ini kalimat yang diperiksa ialah kalimat sederhana, kalimat bertingkat, dan kalimat kompleks. Kedua, sistem dapat memberikan pesan kesalahan kepada pengguna sehingga pengguna dapat mengetahui letak kesalahan pada kalimat yang diproses. Ketiga, sistem dapat menampilkan struktur pembentukan sebuah kalimat berdasarkan frase pembentuknya dengan menggunakan kaidah tata bahasa Sakri melalui pohon sintaks. Keempat, sistem mudah untuk dikembangkan menjadi pemeriksaan semantis kalimat Bahasa Indonesia, dengan cara mengelompokkan kata-kata dalam *database* sistem lebih spesifik lagi dan aturan-aturan dalam *database* sistem juga diubah sesuai dengan simbol yang baru. Kelima, dengan adanya penambahan simbol yang disesuaikan dengan golongan kata, maka aturan-aturan yang sudah ada dapat diperbarui dan diperbanyak untuk disesuaikan kaidah tata bahasa secara semantis.

Hal-hal yang belum mampu dijawab oleh aplikasi ini, antara lain: (1) tanda koma selain berfungsi sebagai tanda hubung, juga dapat diartikan sebagai penanda intonasi. Jika sistem menemukan tanda koma, maka akan dianggap sebagai tanda hubung; (2) sistem belum dapat mengenali kalimat kompleks secara sempurna. Hal ini karena kalimat kompleks dapat terdiri dari lebih dari satu jenis kalimat di antaranya kalimat setara; (3) sistem belum dapat memproses kata yang memiliki jenis kata lebih dari satu dan jenis kata hubung yang mengapit kata atau frase; (4) struktur pembentukan setiap frase hampir serupa, sehingga sulit untuk menentukan akhir dari sebuah frase, di sini diperlukan peran semantik untuk menunjukkan akhir dari sebuah frase.

Tabel 2

Hasil Pengujian Terhadap Kalimat yang Benar berdasarkan Sakri

Jenis Kalimat	Jumlah Kalimat	Persentase Benar(%)
Kalimat Tunggal	32	96,88
Kalimat Bertingkat	104	99.04
Kalimat Kompleks	59	91,53

Tabel 3

Hasil Pengujian Terhadap Kalimat yang Tidak Valid Berdasarkan Sakri

Jumlah Kalimat	Persentase Benar(%)
14	100

PENUTUP

Simpulan

Dari hasil penelitian dapat disimpulkan (1) secara keseluruhan keberhasilan sistem dalam memeriksa kesalahan sintaksis pada kalimat sederhana, kalimat bertingkat, dan kalimat kompleks berdasarkan kaidah tata bahasa yang disusun oleh Sakri cukup tinggi, yaitu di atas 91%; (2) sistem telah dapat memberikan informasi kepada pengguna mengenai kalimat yang sudah memenuhi kaidah tata Bahasa Indonesia atau kalimat yang belum memenuhi kaidah tata bahasa Indonesia. Juga memungkinkan pengguna untuk memperbaiki kalimat yang masih belum sesuai dengan kaidah tata bahasa Indonesia; (3) sistem belum dapat mengenali kalimat kompleks secara sempurna; (4) sistem belum dapat memproses kata yang memiliki jenis kata lebih dari satu dan jenis kata hubung yang mengapit kata atau frase; (5) sistem belum mampu membedakan fungsi tanda koma sebagai tanda hubung atau penanda intonasi.

Saran

Perancangan sistem ini dapat dijadikan sebagai dasar bagi sistem penganalisis kalimat yang dapat menganalisis secara semantis, tetapi harus diadakan penelitian lebih lanjut. Beberapa saran untuk pengembangan lebih lanjut seperti (1) jumlah kata yang terdapat dalam kamus kata diperbanyak; (2) mengembangkan kemampuan analisis sistem, sehingga dapat memeriksa kalimat-kalimat kompleks; (3) untuk kata-kata yang ada dapat diberi simbol yang lebih spesifik yang sudah mengacu pada golongan katanya; (4) mengembangkan sistem sehingga dapat membedakan tanda koma antara sebagai tanda hubung atau penanda intonasi kalimat. Dalam proses pengenalan koma ini diperlukan peran semantis; (5) untuk kasus kalimat setara, dapat diatasi dengan pengembangan semantisnya. Apakah kata-kata yang sebelum dan sesudah tanda hubung tersebut merupakan suatu kalimat yang bermakna atau bukan. Jika benar, maka kalimat tersebut dapat diproses sebagai dua kalimat yang terpisah.

DAFTAR PUSTAKA

Aho, A.V., Lam, M.S.,Sethi, R., Ullman, J.D. (2007). *Compiler Principles, Techniques and Tools*. Boston: Addison-Wesley.

- Arman, Arry Akhmad (2005). Characteristics of Indonesian Language and Development of Human Language Technology in Indonesia. *Proceeding Oriental-COCOSDA 2005*. Jakarta: BPPT.
- Badan Pengkajian dan Penerapan Teknologi. (2004). *KEBI Online: The Indonesian Electronic Dictionary Online*. Diakses dari <http://www.nlp.aia.bppt.go.id/kebi>.
- Departemen Pendidikan dan Kebudayaan. (1990). *Kamus Besar Bahasa Indonesia*. Jakarta: Balai Pustaka.
- Mandala, R., Sigit, A., Munir, R., Harlili (2004). Sistem Penganalisis Sintaksis Otomatis dengan Metode Generalized-LR Parsing. *Proceeding Seminar Nasional Aplikasi Teknologi Informasi*. Bandung: ITB.
- Pressman, Roger S. (2001). *Software Engineering : A Practitioner's Approach*, (5th ed.). New York: McGraw-Hill.
- Raharjo, Budi (2001). *Apa itu Regular Expression?* Diakses dari <http://budi.insan.co.id/articles/liburan-2001.html>.
- Sakri, Adapt. (1994). *Bangun Kalimat Bahasa Indonesia*, (ed. 2). Bandung: ITB.